

Tomáš KOT^{*}, Vladimír MOSTÝN^{}, Petr NOVÁK^{***}**

**APPLICATION OF A 3D REPRESENTATION OF 2D LASER SCANNER DATA
IN OPERATOR CONTROLLED MOBILE ROBOTS**

**VYUŽITÍ 3D ZNÁZORNĚNÍ DAT 2D LASEROVÉHO SCANNERU
PRO MOBILNÍ ROBOTY ŘÍZENÉ OPERÁTOREM**

Abstract

A method of 3D visualization of 2D laser scanner distance data is described in this paper. The method is intended as an aid for human operator controlling a mobile robot, as a complement to camera subsystem. The system uses a single 2D laser scanner and renders its data in the form of interconnected 3D rectangles together with a simplified model of the mobile robot, using perspective projection. The created application is programmed in Visual C++; graphics is rendered using Microsoft Direct3D.

Abstrakt

Článek popisuje metodu 3D vizualizace 2D délkových dat laserového scanneru. Cílem metody je pomoci lidskému operátorovi při řízení mobilního robotu, jedná se o doplněk kamerového subsystému. Systém využívá jeden 2D laserový scanner a vykresluje jeho data v podobě vzájemně navazujících 3D obdélníků společně se zjednodušeným modelem mobilního robotu s využitím perspektivního promítání. Vytvořená aplikace je naprogramována ve Visual C++, grafika je renderována s využitím Microsoft Direct3D.

1 INTRODUCTION

2D laser scanners are being used not only as monitoring and safety devices in industry, but still more and more also in service robotics. Here, these sensors find their use usually in autonomous mobile robots that are able to build a map of the surroundings based on a set of scans and use it for its own navigation.

In many situations, however, mobile robots are controlled remotely by a human operator, who still has a bigger decision-making ability. This concerns mainly mobile robotics in safety engineering, for example reconnaissance or emergency robots employed during fires, chemical contaminations etc. The operator has video from the camera subsystem at the disposal, which usually comprise one or two pivoted cameras. Even with the usage of multiple cameras with rotation, quick navigation in unknown confined places is quite difficult because of the very limited angle of view of cameras (usually between 40 and 60 degrees, with special lenses up to 100 degrees).

A laser scanner can provide additional helpful real-time information about the surroundings, in the angle significantly larger than a camera's field of view (FOV) and even with display of the actual distance values to obstacles. It is however important to choose a clear and obvious representation of these data.

^{*} Ing., Department of Robotics, Faculty of Mechanical Engineering, VŠB - Technical University of Ostrava, 17. listopadu 15, Ostrava, tel. (+420) 597 329 363, e-mail tomas.kot@vsb.cz

^{**} prof. Dr. Ing., Department of Robotics, Faculty of Mechanical Engineering, VŠB - Technical University of Ostrava, 17. listopadu 15, Ostrava, tel. (+420) 597 324 257, e-mail vladimir.mostyn@vsb.cz

^{***} doc. Dr. Ing., Department of Robotics, Faculty of Mechanical Engineering, VŠB - Technical University of Ostrava, 17. listopadu 15, Ostrava, tel. (+420) 597 323 595, e-mail petr.novak@vsb.cz

2 LASER SCANNER PRINCIPLE OF OPERATION

A 2D laser scanner [1] provides a set of distance measurements in the sensor's working plane. The sensor finds the distance of a surface of an obstacle by measuring the time a pulse of light needs to reach the surface, reflect and travel back to the sensor. Provided the laser emitter or an additional system of mirrors rotate, the sensor can scan its entire field of view given by the angle of view and maximal distance of the particular type of scanner.

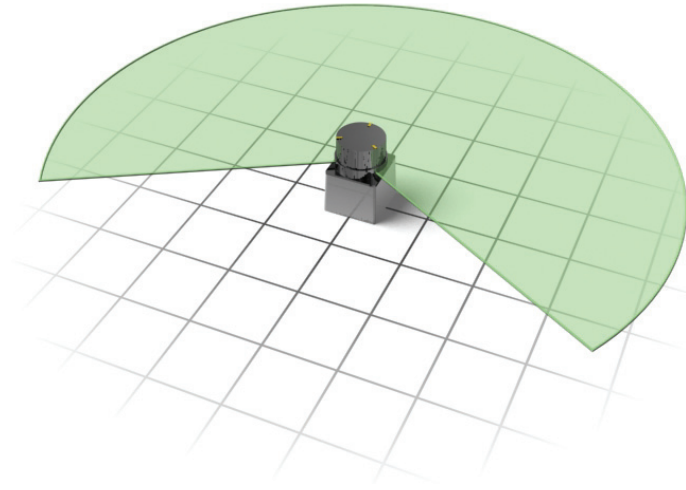


Fig. 1 Working area of a 2D laser scanner with 240° angle of view

The laser scanner HOKUYO URG-04LX (Fig. 2) was used to practically test the result of this research work. In spite of the very small size and low weight (only 160 g) of this sensor, it offers quite good technical parameters [2]:

- ☐ 240° measuring area,
- ☐ angular resolution 0.36°,
- ☐ maximal distance 4095 mm,
- ☐ accuracy ± 10 mm (60 to 1000 mm); 1 % (above 1000 mm),
- ☐ 100 ms scanning time.



Fig. 2 HOKUYO URG-04LX laser scanner

The scanner can be connected to the computer via RS232 or USB line. SDK tools are available [2] to simplify programming of a user application for data acquisition.

3 THREE-DIMENSIONAL DISPLAY OF 2D SCANNER DATA

The laser scanner measurement data are usually displayed in a form similar to the one shown on Fig. 3. That means – corresponding to their nature – strictly in 2D. This representation is however quite confusing and not very clear for human operator and is not always beneficial for navigation.

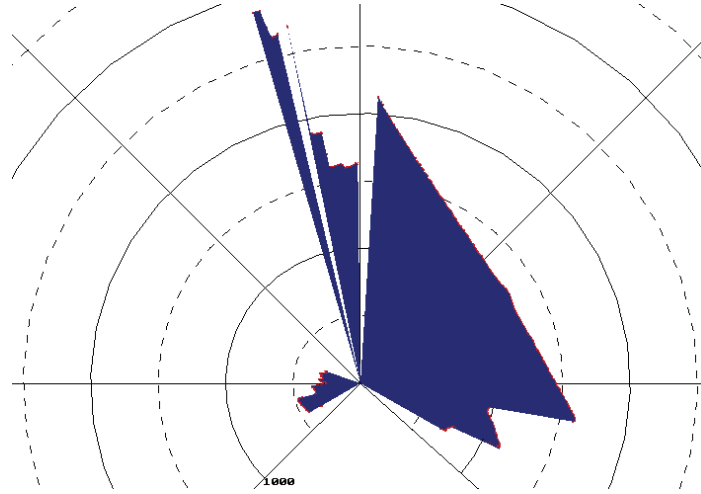


Fig. 3 A typical display of laser scanner measurement data

3D perspective projection (Fig. 4) proved to be much better, because it is a display method natural for human and thus much easier comprehensible.

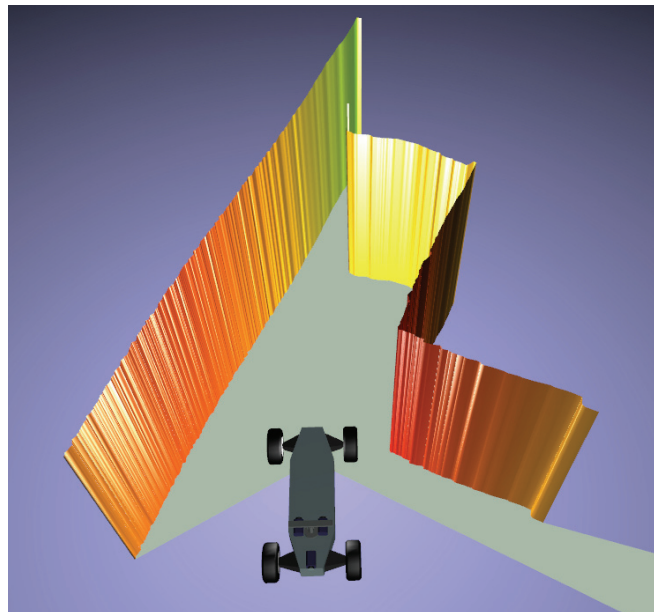


Fig. 4 Three-dimensional perspective projection of 2D scanner data

3.1 Principle of 3D representation

With only one fixed 2D scanner, there is not enough data for a full 3D representation of the scene. The scanner provides only a set of distances measured in one plane (in the form of polar coor-

dinates) and it is not possible to determine the actual height of an obstacle or its shape, if it changes with height.

For navigation, however, this kind of information is not crucial. Provided the scanner is situated around the height of the widest part of the robot, the data are sufficient to decide whether the robot will pass an obstacle or not.

To make a simplified 3D representation of the data, it is possible to make an assumption that all the obstacles are of the same height and have constant shape along the vertical axis. This height thus has no informative value; it works only to increase the clearness of the representation.

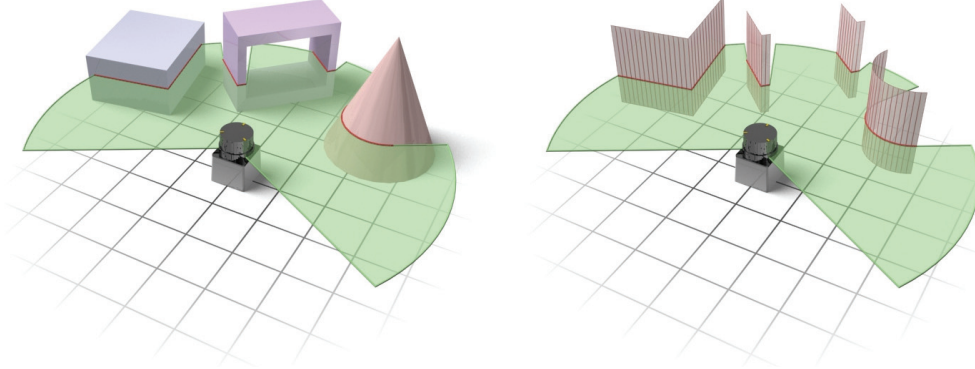


Fig. 5 Example of a scan of different objects and their simplified 3D representation

The URG-04LX scanner measures 681 distance values in the 240° area, which means 681 points represented by polar coordinates. For a 3D display in the form of surfaces, it is necessary to connect each two adjoined points by an imaginary line and extrude it along the vertical axis by a chosen constant distance, which results in a maximum of 680 interconnected rectangular surfaces.

The procedure of calculation of needed values is shown on Fig. 6. Point A_i defined by the corresponding angle φ_i and the measured distance r_i represents the i -th value from the total of $n = 681$. Output from the scanner consists only of distance values r_0 to r_{680} , therefore it is necessary to compute also the angle:

$$\varphi_i = \varphi_0 + \alpha_R \cdot i, \quad (1)$$

$$\varphi_0 = -(\alpha_{MAX} - \pi) / 2, \quad (2)$$

$$\varphi_R = \alpha_{MAX} / (n - 1), \quad (3)$$

where φ_0 is the starting angle, α_R is the angular resolution of the scanner, α_{MAX} is the field of view (precisely 239.77° for URG-04LX) and n is the number of distance values (681).

The following step is conversion from polar coordinates to the Cartesian coordinates:

$$x_i = r_i \cdot \cos(\varphi_i), \quad (4)$$

$$y_i = r_i \cdot \sin(\varphi_i). \quad (5)$$

These coordinates determine point A_i located in the x - y plane, whilst point B_i is shifted along the z axis by the chosen value v :

$$A_i = [x_i, y_i, 0], \quad (6)$$

$$B_i = [x_i, y_i, v]. \quad (7)$$

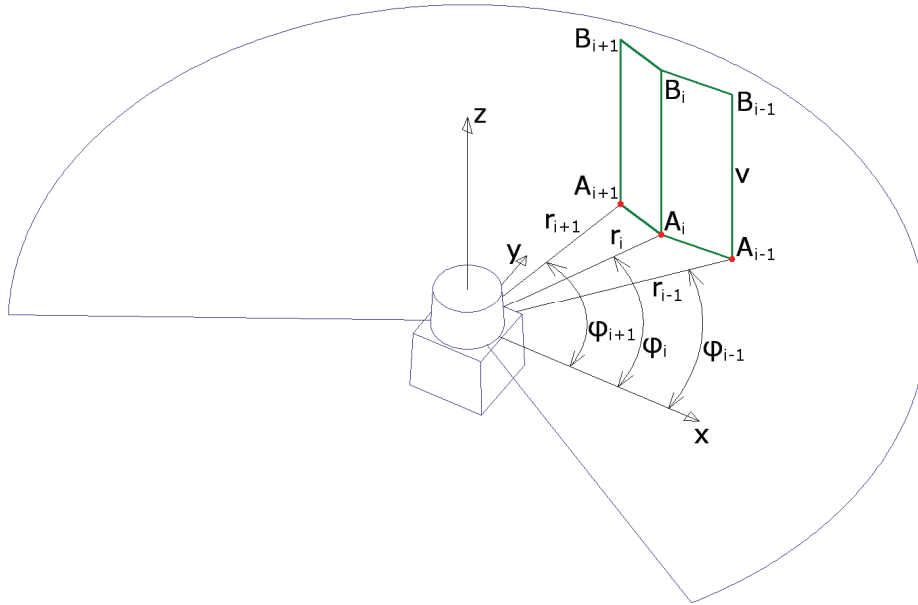


Fig. 6 Definition of rectangular surfaces

3.2 Real-time rendering of 3D graphics

The software application capturing data from the laser scanner and rendering them in the above described way was programmed in the Visual C++ environment and uses the Direct3D API in version 9.0c for graphics rendering. Direct3D uses hardware acceleration and as the complexity of 3D graphics in the application is low, it runs very fast even on systems with a low-end graphics card. With the tested scanner (URG-04LX) which is capable of only 10 scans in a second, the graphics is not the bottleneck of the application.

The application renders, besides the reconstructed 3D data, also the classic 2D representation (in gray color) and a simplified model of the mobile robot. With the robot drawn as well, the operator is able to understand the situation better; he can see easily where the robot is heading, how far the obstacles are and where the robot will probably fit. The distances are marked by a color gradient; red color represents minimal, yellow middle and green maximal distance in the scanner range. The view can be freely rotated and zoomed, so it is possible to view the scene quickly from any angle.

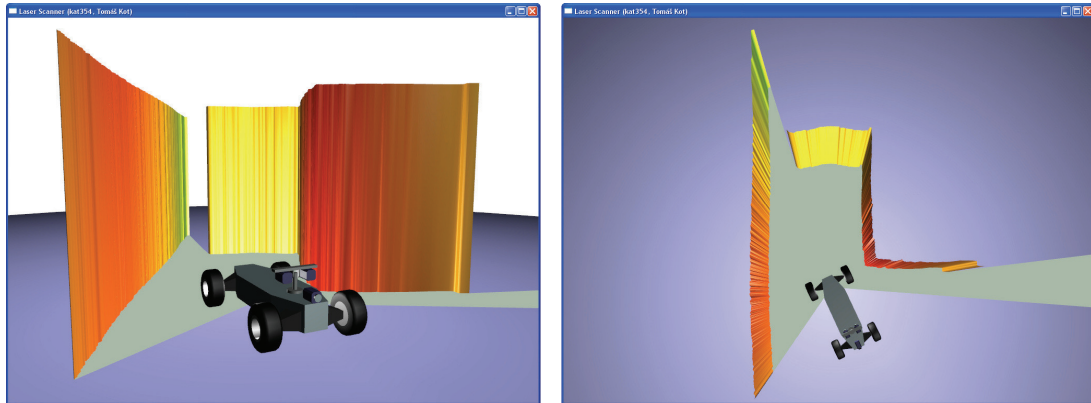


Fig. 7 Different views on the same scene based on real scanned data

The basic renderable surface element in 3D graphics is a triangle unambiguously defined by three points (*vertices*) in space. Each vertex in our case consists of a three-element position vector, a three-element normal vector (necessary for dynamic lighting) and a color (based on the distance). By a connection of two coplanar triangles it is possible to draw a rectangle – and rectangles are the objects drawn in this application; see Fig. 7, where the rectangles $A_{i-1}A_iB_iB_{i-1}$ and $A_iA_{i+1}B_{i+1}B_i$ are shown.

For correct rendering, every rectangle has its own set of 4 vertices, which means there is a total of 4 times 680 vertices. Drawing is done by triangles – two triangles for every rectangle, but the two vertices on diagonal are shared between the triangles, thus it is advantageous to use *indices*, which refer to vertices by their order number. The total number of indices is 6 times 680. Fig. 8 shows the first two rectangles corresponding to the first three scanner distance values; the numbers on the figure represents order number of vertices. The sequence of indices needed to draw these 2 rectangles (4 triangles) is:

$$0, 1, 2; 2, 1, 3; 4, 5, 6; 6, 5, 7. \quad (8)$$

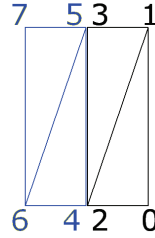


Fig. 8 Schematic representation of vertices necessary for rendering of two adjoining rectangles

The vertices must be updated for every scan. This updating consists of calculation of the new position, normal and color. The position vector is calculated according to equations (1) to (7) and the color is linearly interpolated between the key colors (red – yellow – green) based on the value r_i . The normal vector is identical for all 4 vertices of one rectangle and is computed to be perpendicular to the rectangle:

$$\mathbf{v}_H = \mathbf{p}_2 - \mathbf{p}_0, \quad (9)$$

$$\mathbf{v}_V = (0; 0; 1), \quad (10)$$

$$\mathbf{n} = \mathbf{v}_H \times \mathbf{v}_V, \quad (11)$$

where \mathbf{p}_i is the position vector of the i -th vertex, \mathbf{v}_V is a unit vertical vector and \mathbf{n} is a normal vector. Vector \mathbf{v}_H is normalized after the step (9), using a built-in Direct3D function.

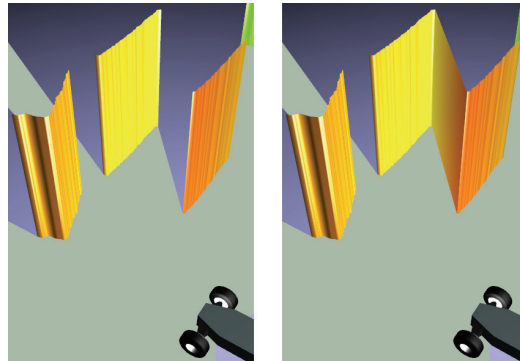


Fig. 9 A rejected rectangle with big difference between two adjoining distance values

As it is evident for example from Fig. 7, in a general case not all the 680 rectangles are drawn; values out of the range of the scanner are ignored and thus only representations of existing obstacles are displayed.

Also rectangles between two adjoined scanner points that have the absolute value of distance difference ($r_{i+1} - r_i$) bigger than a chosen limit are rejected. This occurs when a nearer object in the scene is shadowing another object. If the join was drawn, a false impression that the objects are connected would be made, as illustrated on Fig. 9, where the picture on the right is drawn without this filtering and the shape of the obstacle is misrepresented. The pictures differs only in one wide rectangle.

The following figures show some confrontations of rendered 3D representations of the scanner data and the corresponding real scenes.

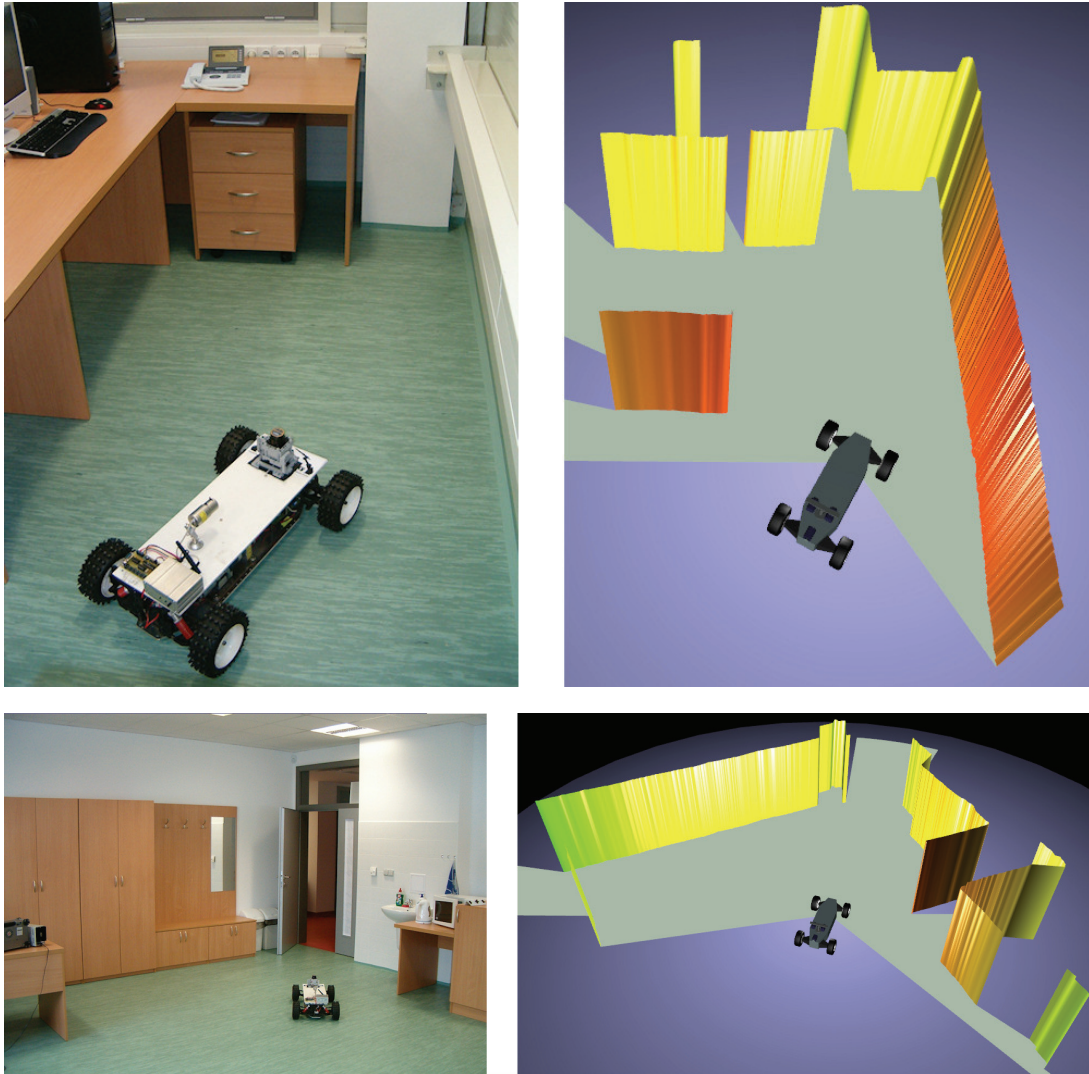


Fig. 10 Photographs of real scenes and the corresponding 3D reconstructions

4 CONCLUSION

The method of 3D representation of 2D scanner data suggested and described in the paper and tested using the created Direct3D application can significantly improve the human operator's ability to navigate a mobile robot in unknown environment, especially when using only a low number of cameras. It can be easily further extended to display also numeric values of distances to prominent points, to automatically warn the operator about possible collisions or even to stop the robot when a collision is about to occur. The system is not intended to replace the camera subsystem in remote mobile robot navigation, but merely to complement it.

The current version of the application runs in real time, with the rate of 10 updates per second, which is a limitation of the used scanner. This speed is on the margin of acceptability for fast moving mobile robots and could be improved by choosing a better scanner. For a real practical application it is also necessary to provide a reliable wireless data transfer with corresponding speed.

This article was compiled as part of project FT-TA5/071, supported by the Industrial Research and Development Program of the Ministry of Industry and Trade.

REFERENCES

- [1] *Laser Range Scanner* [online]. 2009. Available from WWW:
<http://en.wikipedia.org/wiki/Laser_Range_Scanner>.
- [2] HOKUYO. *Scanning Range Finder URG-04LX* [online]. 2008. Available from WWW:
<http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html>.
- [3] MICROSOFT. *Direct3D 9 Graphics* [online]. 2009. Available from WWW:
<[http://msdn.microsoft.com/cs-cz/library/bb219837\(en-us,VS.85\).aspx](http://msdn.microsoft.com/cs-cz/library/bb219837(en-us,VS.85).aspx)>.